



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

MN

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/829,205	04/22/2004	Masato Mitsumori	NIT-421	5534
24956	7590	04/16/2007	EXAMINER	
MATTINGLY, STANGER, MALUR & BRUNDIDGE, P.C. 1800 DIAGONAL ROAD SUITE 370 ALEXANDRIA, VA 22314			KROFCHECK, MICHAEL C	
			ART UNIT	PAPER NUMBER
			2186	
SHORTENED STATUTORY PERIOD OF RESPONSE	MAIL DATE	DELIVERY MODE		
3 MONTHS	04/16/2007	PAPER		

Please find below and/or attached an Office communication concerning this application or proceeding.

If NO period for reply is specified above, the maximum statutory period will apply and will expire 6 MONTHS from the mailing date of this communication.

Office Action Summary	Application No.	Applicant(s)	
	10/829,205	MITSUMORI ET AL.	
	Examiner	Art Unit	
	Michael Kofcheck	2186	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --
Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1) Responsive to communication(s) filed on 29 January 2007.
 2a) This action is **FINAL**. 2b) This action is non-final.
 3) Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4) Claim(s) 1-4,6-10 and 12-15 is/are pending in the application.
 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
 5) Claim(s) _____ is/are allowed.
 6) Claim(s) 1-4,6-10 and 12-15 is/are rejected.
 7) Claim(s) _____ is/are objected to.
 8) Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9) The specification is objected to by the Examiner.
 10) The drawing(s) filed on 22 April 2004 is/are: a) accepted or b) objected to by the Examiner.
 Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
 Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
 11) The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12) Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
 a) All b) Some * c) None of:
 1. Certified copies of the priority documents have been received.
 2. Certified copies of the priority documents have been received in Application No. _____.
 3. Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) Notice of References Cited (PTO-892)
 2) Notice of Draftsperson's Patent Drawing Review (PTO-948)
 3) Information Disclosure Statement(s) (PTO/SB/08)
 Paper No(s)/Mail Date 1/29/2007.

4) Interview Summary (PTO-413)
 Paper No(s)/Mail Date. _____.
 5) Notice of Informal Patent Application
 6) Other: _____.

DETAILED ACTION

1. This office action is in response to the amendment filed on 1/29/2007.
2. Claims 1, 4, 7-10, 12-13 have been amended.
3. Claims 5 and 11 have been cancelled.
4. The objections/rejections from the prior correspondence not restated herein have been withdrawn.

Claim Objections

5. Claims 13-15 are objected to because of the following informalities:
 - a. With respect to claim 13, the amendment to the claim reads, ..."wherein the first program code and the second program code have such and interrelationship that the first program code is a program code invoked by the first program code..." Independent claims 1 and 7 indicate that the first program code invokes the second, not the first invoking itself. Appropriate correction is required.

Claim Rejections - 35 USC § 103

6. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

7. The factual inquiries set forth in *Graham v. John Deere Co.*, 383 U.S. 1, 148 USPQ 459 (1966), that are applied for establishing a background for determining obviousness under 35 U.S.C. 103(a) are summarized as follows:

1. Determining the scope and contents of the prior art.
2. Ascertaining the differences between the prior art and the claims at issue.
3. Resolving the level of ordinary skill in the pertinent art.
4. Considering objective evidence present in the application indicating obviousness or nonobviousness.

8. This application currently names joint inventors. In considering patentability of the claims under 35 U.S.C. 103(a), the examiner presumes that the subject matter of the various claims was commonly owned at the time any inventions covered therein were made absent any evidence to the contrary. Applicant is advised of the obligation under 37 CFR 1.56 to point out the inventor and invention dates of each claim that was not commonly owned at the time a later invention was made in order for the examiner to consider the applicability of 35 U.S.C. 103(c) and potential 35 U.S.C. 102(e), (f) or (g) prior art under 35 U.S.C. 103(a).

9. Claims 1-2, 7-8, 13-14 rejected under 35 U.S.C. 103(a) as being unpatentable over the applicant's admitted prior art (AAPA), Hardin et al., US patent application publication 2002/0161961, and Gee et al., US patent 6374286 (which is incorporated by reference into Hardin, paragraph 23 of Hardin).

10. With respect to claim 1, AAPA teaches of a method of detecting invalid memory access used in a computer which executes a language system having a specific memory management function; a first program code that is executed under the control of the language system, and that accesses a first memory area reserved by the

Art Unit: 2186

language system; and a second program code that is directly executed under the control of OS, and that accesses a second memory area reserved by the OS (applicant's specification page 1, line 15-page 2, line 6);

AAPA fails to explicitly teach of wherein said method executed by the language system detects invalid memory access to the first memory area caused by the second program code.

Hardin teaches of wherein said method executed by the language system detects invalid memory access to the first memory area caused by the second program code, said method executed by the language system comprising the steps of: setting the memory protection of the first memory area before the first program code calls the second program code (fig. 5; paragraph 0036-0038; it is abundantly clear to one of ordinary skill in the art that the MVM control enabled the memory protection before calling the second VM. Failing to do this would cause a period of time where the second VM could modify the memory that should be protected);

when a memory protection exception occurs, notifying of invalid memory access caused by the second program code to outside (paragraph 0056-0057); and

when the execution of the second program code ends and the control returns to the language system, disabling the memory protection of the first memory area (it is abundantly clear to one of ordinary skill in the art to disable the memory protection, upon the completion of the second VM, as the threat of the memory area being overwritten by another VM no longer exists).

Gee teaches of wherein the first program code and the second program code have such an interrelationship that the first program code is a calling program while the second program code is a program code invoked by the first program code (column 23, lines 23-40; where the master JVM (executive code) starts a proxy thread to run the next JVM. Thus the master JVM calls the next JVM via the proxy thread, and the next JVM is invoked by the master JVM)

calling and executing the second program code from the first program code (column 23, lines 23-40; where the master JVM (executive code) starts a proxy thread to run the next JVM. Thus the master JVM calls the next JVM via the proxy thread, and the next JVM is invoked by the master JVM).

It would have been obvious to one of ordinary skill in the art having the teachings of the applicant's admitted prior art, Hardin, and Gee at the time of the invention to incorporate the memory protection as taught in Hardin (incorporating Gee) into AAPA. Their motivation would have been to run multiple virtual machines/Operating Systems (Hardin, paragraph 0003).

11. With respect to claim 7, the combination of AAPA, Hardin, and Gee teach of the limitations cited above with respect to claim 1. Additionally, since the limitations are taught in the context of a computer system, there must be a program on a readable medium that causes its functionality.

12. With respect to claims 2 and 8, the combination of AAPA, Hardin, and Gee teaches of wherein: when said memory protection exception occurs, if it is detected that the first program code performs normal memory access to the first memory area, said

language system disables the memory protection to allow the normal memory access, and then enables the memory protection again (fig. 5, 6; paragraph 0036-0038, 0040; where in response to the first VM accessing an address in its area, the first VM is allowed to due such by the MVM control).

13. With respect to claim 13, AAPA teaches of a language system used in a computer which executes a language system having a specific memory management function; a first program code that is executed under the control of the language system, and that accesses a first memory area reserved by the language system; and a second program code that is directly executed under the control of OS, and that accesses a second memory area reserved by the OS (applicant's specification page 1, line 15-page 2, line 6);

Hardin teaches of wherein said language system detects invalid memory access to the first memory area caused by the second program code, said language system comprising: means for setting memory protection of the first memory area before the first program code calls the second program code (fig. 5; paragraph 0036-0038; it is abundantly clear to one of ordinary skill in the art that the MVM control enabled the memory protection before calling the second VM. Failing to do this would cause a period of time where the second VM could modify the memory that should be protected),

for notifying of invalid memory access caused by the second program code to outside when a memory protection exception occurs (paragraph 0056-0057); and

means for disabling the memory protection when the execution of the second program code ends and the control returns to the language system (it is abundantly clear to one of ordinary skill in the art to disable the memory protection, upon the completion of the second VM, as the threat of the memory area being overwritten by another VM no longer exists. This is done by the MVM control).

Gee teaches of wherein the first program code and the second program code have such an interrelationship that the first program code is a program code invoked by the first program code (column 23, lines 23-40; where the master JVM (executive code) starts a proxy thread to run the next JVM. Thus the master JVM calls the next JVM via the proxy thread, and the next JVM is invoked by the master JVM)

for calling and executing the second program code from the first program code (column 23, lines 23-40; where the master JVM (executive code) starts a proxy thread to run the next JVM. Thus the master JVM calls the next JVM via the proxy thread, and the next JVM is invoked by the master JVM).

14. With respect to claim 14, the combination of AAPA, Hardin, and Gee teaches of means, when said memory protection exception occurs, if it is detected that the first program code performs normal memory access to the first memory area, for disabling the memory protection, allowing the normal memory access, and then enabling the memory protection again (fig. 5, 6; paragraph 0036-0038, 0040; where in response to the first VM accessing an address in its area, the first VM is allowed to due such by the MVM management).

Art Unit: 2186

15. Claims 4, 10, rejected under 35 U.S.C. 103(a) as being unpatentable over AAPA, Rinne et al., US patent 6098194, Hardin, and Gee (incorporated into Hardin).

16. With respect to claim 4, AAPA teaches of a method of detecting invalid memory access used in a computer which executes a language system having a specific memory management function; a first program code that is executed under the control of the language system, and that accesses a first memory area reserved by the language system; and a second program code that is directly executed under the control of OS, and that accesses a second memory area reserved by the OS (applicant's specification page 1, line 15-page 2, line 6);

AAPA fails to explicitly teach of wherein said method executed by the language system detects invalid memory access to the first memory area caused by the second program code.

The combination of AAPA and Rinne teaches of wherein said method executed by the language system detects invalid memory access to the first memory area caused by the second program code, said method executed by the language system comprising the steps of: allowing said language system to save code information associated with the contents of the first memory area before the first program code calls the second program code (Rinne, fig. 2; column 35-38, column 5, lines 47-58; where a checksum is calculated over a specific storage area. As it is used later, it must be stored. In the combination of AAPA and Rinne, it is done prior to the execution of the other language program);

when the execution of the second program code ends and the control returns to the language system, judging whether or not code information associated with the contents of the first memory area coincides with the saved code information; and if the code information associated with the contents of the first memory area does not coincide with the saved code information, notifying of invalid memory access caused by the second program code to outside (Rinne, fig. 2; column 35-38, column 5, lines 47-58; where the checksum is recalculated and if it is not the same an error message is produced. In the combination this is done upon the completion of the other language program to determine if an invalid memory access occurred).

Gee teaches of calling and executing the second program code from the first program code (column 23, lines 23-40; where the master JVM (executive code) starts a proxy thread to run the next JVM. Thus the master JVM calls the next JVM via the proxy thread, and the next JVM is invoked by the master JVM);

The combination of AAPA, Rinne, Hardin, and Gee teach of wherein, when it is detected that while the second program code is called the first program code normally updates the first memory area, said language system updates saved code information based on the code information associated with contents of the first memory area updated (Rinne, fig. 3, column 2, lines 45-51; where when the memory contents are changed, the old portion of the checksum is deleted and a new portion is added based on the changes contents; this occurs as a result of any allowed write access to the memory at any time).

It would have been obvious to one of ordinary skill in the art having the teachings of AAPA and Rinne at the time of the invention to include the verifying the contents of a storage area via a checksum comparison as taught in Rinne in AAPA. Their motivation would have been to determine if specific storage areas are free from defects so they do not have to be re-loaded (Rinne, Column 5, lines 35-42).

It would have been obvious to one of ordinary skill in the art having the teachings of the applicant's admitted prior art, Rinne, Hardin, and Gee at the time of the invention to incorporate the memory protection as taught in Hardin (incorporating Gee) into AAPA. Their motivation would have been to run multiple virtual machines/Operating Systems (Hardin, paragraph 0003).

17. With respect to claim 10, the combination of AAPA, Rinne, Hardin, and Gee teach of the limitations cited above with respect to claim 4. Additionally, since the limitations are taught in the context of a computer system, there must be a program that causes its functionality.

18. Claims 3, 9, and 15 are rejected under 35 U.S.C. 103(a) as being unpatentable over AAPA, Hardin and Gee as applied to claims 1, 7, and 13 respectively, and further in view of Wenocur et al., US patent application publication 2002/0165912.

19. With respect to claims 3, 9, and 15, Wenocur teaches of wherein: if the first program code is executed under the multithread control, said language system suspends the execution of other threads while a certain thread calls the second program code (paragraph 1061).

It would have been obvious to one of ordinary skill in the art having the teachings of AAPA, Hardin, Gee and Wenocur at the time of the invention to include the process of switching threads as taught in Wenocur in the combination of AAPA, Hardin, and Gee. Their motivation would have been to optimize processor time by switching active threads.

20. Claims 6 and 12 are rejected under 35 U.S.C. 103(a) as being unpatentable over AAPA, Rinne, Hardin, and Gee as applied to claims 4 and 10 respectively, and further in view of Wenocur.

21. With respect to claims 6 and 12, Wenocur teaches of wherein: if the first program code is executed under the multithread control, said language system suspends the execution of other threads while a certain thread calls the second program code (paragraph 1061).

It would have been obvious to one of ordinary skill in the art having the teachings of AAPA, Rinne, Hardin, Gee, and Wenocur at the time of the invention to include the process of switching threads as taught in Wenocur in the combination of AAPA, Rinne, Hardin, and Gee. Their motivation would have been to optimize processor time by switching active threads.

Response to Arguments

22. Applicant's arguments filed on 1/29/2007 have been fully considered but they are not persuasive.

23. Applicant argues with respect to claims 1-2, 7-8, and 13-14 that Hardin fails to teach of the second program code being directly executed under control of an OS. The examiner would like to point out that this is taught in the AAPA, as mentioned in paragraph 15 the office action mailed on 9/28/2006 and in paragraph 10 of this correspondence. Specifically, lines 22-25 on page 1 of the applicant's specification.

24. Applicant's arguments regarding claims 4 and 10 do not comply with 37 CFR 1.111(c) because they do not clearly point out the patentable novelty which he or she thinks the claims present *in view of the state of the art disclosed by the references cited or the objections made*. Further, they do not show how the claim limitations avoid such references or objections.

The examiner is unclear what the applicant is arguing regarding ruling out monitoring the storage area as the first program code updates the storage area during the monitoring. There is no such limitation in the claim.

The applicant alleges that Rinne does not teach of when it is detected that while the second program code is called, the first program code normally updates the first memory area, the saved code information is updated based on code information associated with contents of the updated first memory area, and does not point out how that claim limitation is avoided in the references.

The examiner disagrees that said claim limitations are not taught by Rinne. The explanation for the rejection is written above in paragraph/section 16.

25. Applicant's arguments with respect to claims 1-2, 7-8, and 13-14 have been considered but are moot in view of the new ground(s) of rejection.

Conclusion

26. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.
27. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).
28. A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.
29. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Michael Kroccheck whose telephone number is 571-272-8193. The examiner can normally be reached on Monday - Friday.
30. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Matt Kim can be reached on 571-272-4182. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

31. Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).



Michael Kofcheck



MATTHEW KIM
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100